REMARKS

Claims 1, and 3-20 are pending. By this amendment, claims 1, 3-5, 8-9, and 12-16 are amended. Claim 2 is canceled. New claims 19-20 are added. No new matter is introduced. Reconsideration and allowance of the claims in view of the above-amendments and the remarks that follow are respectfully requested.

Specification

The specification has been objected to for two reasons, namely the inclusion of a non-descriptive title and the inconsistent use of "op6" and "op8" on pages 8 and 9. The title has been amended to more particularly describe the invention as claimed. The specification has also been amended as suggested by the examiner to clarify the use of "op6" and "op8" on pages 8 and 9. The specification is now in a proper form. Reconsideration of the objections to the specification is requested.

Drawings

Figure 3 of the drawings has been objected to for failing to label the MXCSR register (130) and the functional units (172, 174). A Proposed Drawing Corrections is filed herewith, including an amended drawing sheet to replace Figure 3, as filed. Reconsideration of the objections to the drawings is requested.

Claims

Claim Objections

Claims 13 and 14 are objected to on the basis that they depend from computer system claim 12, yet recite "The method of claim 12…" (emphasis added). Claims 13 and 14 have been amended to replace "method" with "computer system" as suggested by the examiner. Claim 16 has been objected to as being informal, for using the term "emulated" where "emulate" is proper. Claim 16 has been amended to correct this informality. Reconsideration of the objections to claims 13, 14, and 16 is requested.

Claim Rejections

Claims 1-8 and 10-17 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. patent no. 6,233,671 B1 to Abdallah et al. (hereinafter "Abdallah1") in view of published U.S. patent no. 6,085,312 to Abdallah et al. (hereinafter "Abdallah2"). Claim 9 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Abdallah1 in view of Abdallah2 and U.S. patent no. 6,038,652 to Phillips (hereinafter "Phillips"). Claim 18 is rejected under 35 U.S.C. § 103(a) as being unpatentable over Abdallah1 in view of Abdallah2 and U.S. patent no. 6,321,327 to Makineni (hereinafter "Makineni"). These rejections are respectfully traversed.

Claim 1

Claim 1 has been amended to recite that all of the microinstructions are issued simultaneously, in parallel, and to incorporate the limitations of canceled claim 2. Support for this amendment is found, for example, at page 9, lines 3-13 of the specification. Unlike the invention recited in amended claim 1, the Abdallah patents describe a staggered, or serial, execution scheme (see, e.g., Abdallah 1, col. 5, l. 58; Abdallah 2, col. 2, l. 61; col. 3, ll. 1-16) using a "back off" register (see, e.g., Abdallah 1, Figure 4A, "M3") to handle exceptions, as described in the "Background" section of the present specification.

Neither Abdallah 1 nor Abdallah 2 teach or suggest issuing all of the microinstructions simultaneously. To the contrary, both Abdallah 1 and Abdallah 2 require delaying execution of the second microinstruction. "An operation specified by the single macro instruction is then performed independently on a first and second plurality of the corresponding data elements from said first and second packed data operands at different times using the same circuit to independently generate a first and second plurality of the resulting data elements." Abdallah 1, col. 2, ll. 32-38 (emphasis added). "The segments are sequentially processed by introducing a delay into the processing of the subsequent segments." Abdallah 1, col. 3, ll. 38-40 (emphasis added). Figure 3 of Abdallah 1 makes clear that the separate microinstructions are processed sequentially in time, in a "staggered execution." See, e.g., Abdallah 1, col. 5, l. 58. Abdallah 2 discusses other aspects of the same "staggered" execution scheme. See, e.g., Abdallah 2, col. 2, l. 61; col. 3, ll. 1-16. Figure 2 of Abdallah 2 corresponds to Figure 3 of Abdallah 1 and discloses the same time-delayed, sequential execution, rather than a "simultaneous" execution of microinstructions as recited in amended claim 1.

Further, it would not be obvious to modify either Abdallah 1 or Abdallah 2 to provide for "simultaneous" execution of all of the microinstructions, because the delay in processing is the very heart of the Abdallah inventions. Abdallah specifically and deliberately inserts this delay to create a staggered execution. This allows the microinstructions to execute sequentially on the same hardware to improve efficiency of use of the hardware in multimedia applications. See, e.g., Abdallah 1, col. 5, ll. 45 – col. 6, l. 4; Abdallah 2, col. 3, ll. 17-49. Because the cited references do not teach or suggest allowing simultaneous execution of the microinstructions, claim 1 as amended should be allowed. Claims 3-9 and 19 depend from claim 1 and are allowable as dependent claims. Reconsideration is requested.

Claims 3-5 and 8

Claims 3-5 and 8 have been amended for reasons unrelated to patentability, to comport with the language of amended claim 1, from which claims 3-5 and 8 depend. Claims 3-5 and 8 should now be allowable. Reconsideration is requested.

Claim 9

Claim 9 has been amended to recite that, "if an unmasked exception occurs," "all of the plurality of microinstructions" are canceled "without regard to the relative ages of each of the plurality of microinstructions." Support for this amendment is found, for example, at page 9, lines 14-25 of the specification. None of the cited references teach or suggest canceling all of the microinstructions whenever an unmasked exception occurs. Abdallah 2 discusses handling of exceptions. It describes a system that delays updating the architectural state of a first microinstruction until the exception status of the second, staggered microinstruction is known, using conventional RISC architecture. See, e.g., Abdallah 2, col. 8, ll. 44-47. Conventional RISC architecture flushes only those operations subsequent to the operation taking an exception; earlier operations are allowed to proceed. Conventional systems either use a "back off" register to temporarily store the result of the first executed microinstruction, or a "back off" "undo" mechanism that commits the result of the first microinstruction, but changes back the architectural state if the second executed microinstruction causes a fault. Abdallah 2 neither discloses nor teaches anything to change this RISC convention. In contrast, the method recited in amended claim 9 cancels all pending microinstructions upon detection of an unmasked exception with any of the microinstructions and does not require a "back off" mechanism. As amended, claim 9 is allowable. Reconsideration is requested.

Claim 10

The rejection of claim 10 is respectfully traversed. Claim 10 is rejected as being obvious over Abdallah 1 in view of Abdallah 2. The office action asserts that because Abdallah 1 and Abdallah 2 discuss execution in parallel, they necessarily teach "simultaneous" execution. "If the instructions were not dispatched simultaneously, they would not be executed in parallel." Office Action, p. 8, ll. 1-2. To the contrary, Abdallah 1 and Abdallah 2 clearly discuss a "sequential" or "staggered" execution scheme that allows the same hardware to be reused for both decomposed microinstructions. The very title of Abdallah 1 makes clear that it describes "staggering execution." For the reasons discussed above with respect to amended claim 1, neither Abdallah 1 or 2 teach or suggest "dispatching the high-half and low-half operations simultaneously" as recited in claim 10. Both Abdallah references specifically require that the execution of microinstructions occur "sequentially" in a "staggered" manner so that the same hardware is used to process both microinstructions.

Further, the rejection of claim 10 is improper because neither Abdallah 1 nor Abdallah 2 disclose or suggest canceling a pending microinstruction in a floating point unit when "either" of the microinstructions causes an exception. As discussed above with respect to amended claim 9, conventional RISC architecture cancels only subsequent microinstructions. For this reason, claim 10 should be allowed.

Respectfully, claim 10 should be allowed as-filed. Claims 11 and 20 depend from claim 10 and is allowable as a dependent claim. Reconsideration is requested.

Claim 11

Claim 11 is also allowable for the reasons discussed above with respect to amended claim 9 and claim 10. Conventional RISC architecture cancels pending microinstructions based upon the order in which they were received. Claim 11 cancels both pending instructions upon detection of a fault in either instruction automatically, without regard to the relative ages of the microinstructions. Claim 11 should be allowed as filed. Reconsideration is requested.

Claim 12

Claim 12 has been amended to recite that all of the microinstructions are issued simultaneously. Support for this amendment is found, for example, at page 9, lines 3-13 of the specification. As discussed above with respect to claims 1 and 10, neither Abdallah 1 nor Abdallah 2 teaches or suggests "issuing all of the plurality of microinstructions simultaneously, in parallel." Instead, both Abdallah references teach issuing the microinstructions "sequentially" by intentionally delaying execution of one of the microinstructions to allow the same hardware to process both microinstructions. Using this "staggered execution" process is critical to the Abdallah inventions, because it provides "a significant cost advantage to hardware manufacturers." Abdallah 1, col. 5, l. 54, 58. Claim 12 as amended is now allowable. Claims 13-18 depend from claim 12 and are allowable as dependent claims. Reconsideration is requested.

Claims 13-16

Claims 13-16 have been amended for reasons unrelated to patentability, to comport with the language of amended claim 12, from which claims 13-16 depend. Claims 13-16 should now be allowable. Reconsideration is requested.

New Claims

New claims 19 and 20 have been added to more particularly claim the invention. Support for claims 19 and 20 is found, for example, in the specification at page 5, lines 5-6 and page 9, lines 4-13. The cited art does not teach or suggest microinstructions that are executed in lockstep with each other. Nor does it teach or suggest canceling the pending microinstructions upon receipt of an exception in either microinstruction, without regard to the microinstructions' relative ages and without using a backoff register or other mechanism to store a result of the microinstruction that does not cause an exception. Claims 19 and 20 should be allowed.

Examiner Interview

A telephonic interview was conducted with examiner David J. Huisman on November 20, 2002. The objections to the specification, the title, Figure 3, and claims 13, 14, and 16 were discussed. The rejection of claims 1, 10, and 12 was also discussed in view of Abdallah 1 and Abdallah 2. Specifically, Abdallah's "sequential" or "staggered" execution scheme was discussed in view of the recitation of "simultaneous" issuance and execution of microinstructions recited in claim 10 and in claims 1 and 12 as amended by this amendment. No agreement was reached as to appropriate claim scope.

CONCLUSION

In view of the above amendments and remarks, Applicant respectfully asserts that the application is in condition for allowance. Prompt reexamination and allowance of claims 1 and 3-20 is respectfully requested.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached pages are captioned "**Version with markings to show changes made**." In addition, a clean copy of the pending claims is attached. The attached claims are captioned "**Pending Claims**."

Respectfully submitted,

Date: **November 25, 2002**

Sean S. Wooden, Reg. No. 43,997
DORSEY & WHITNEY LLP
1001 Pennsylvania Ave., N.W.
Suite 400 South
Washington, D.C. 20004
Tel. (202) 442-3000
Fax (202) 442-3199

Attachment:

VERSION WITH MARKINGS TO SHOW CHANGES MADE

In the Specification:

The title has been changed/amended as follows:

METHOD AND COMPUTER SYSTEM FOR DECOMPOSING MACROINSTRUCTIONS INTO MICROINSTRUCTIONS AND EXECUTING THE MICROINSTRUCTIONS SIMULTANEOUSLY [METHOD AND APPARATUS FOR EMULATING SIMD INSTRUCTIONS]

Paragraph beginning at line 26 of page 8 has been amended as follows:

Op1 is the oldest instruction, and if it takes a fault, the processor must flush all younger operations (op2 - opN). If the processor executes two operations in parallel and pipelines new operations every cycle, all these operations might be "in flight" at the same time. For example, op1 and op2 might have issued in parallel and may be close to completion, while op3 and op4 are in flight one cycle behind, op5 and op6 are in flight two cycles behind, etc. If op1 takes an exception, all operations (op1 through [op8] op6) must be flushed. But if op2 takes an exception, only operations op2 through [op8] op6 must be flushed.

In the Claims:

1. (Amended) A method for processing software instructions comprising,

[(a)] decomposing a macroinstruction into a plurality of microinstructions,

[(b)] issuing all [at least two] of the plurality of microinstructions simultaneously, in parallel,

executing all of the plurality of microinstructions simultaneously, in lockstep,

[(c)] determining whether an exception occurs in any of the [at least two of a plurality of] microinstructions, and

[(d)] if an exception occurs in any of the [at least two of a plurality of] microinstructions, canceling all of the [the at least two of a plurality of] microinstructions.

3. (Amended) The method of claim 1[2], wherein the [at least two of a plurality of] microinstructions are executed on separate execution units, but appear as though they were executed on a single execution unit.

4. (Amended) The method of claim 1, wherein all of the [at least two of a plurality of] microinstructions are executed on the same clock cycle.

5. (Amended) The method of claim 1, wherein the [at least two of a plurality of] microinstructions are executed over multiple clock cycles.

8. (Amended) The method of claim 1, further comprising updating a flag based upon a result of the execution of the [at least two of a plurality of] microinstructions.

9. (Amended) The method of claim 1, further comprising,

[(a)] if an unmasked exception occurs, canceling the execution of all of the plurality of microinstructions, without regard to the relative ages of each of the plurality of microinstructions, and invoking a microcode handler, and

[(b)] if an unmasked exception does not occur, updating at least one exception flag by independently generating a logical OR of exceptions for a plurality of functional units.

12. (Amended) A computer system comprising,

a processor comprising,

(a) a floating point unit;

(b) a ROM;

(c) a plurality of floating point registers;

wherein the processor is configured to emulate an instruction set by:

(a) decomposing a macroinstruction into a plurality of microinstructions;

(b) issuing all [at least two] of the plurality of microinstructions simultaneously, in parallel,

(c) determining whether an exception occurs in any of the [at least two of a plurality of] microinstructions, and

(d) if an exception occurs in any of the [at least two of a plurality of] microinstructions, canceling all of the [the at least two of a plurality of] microinstructions.

13. (Amended) The computer system [method] of claim 12, [further comprising] wherein the processor is further configured to emulate the instruction set by executing all of the [at least two of the plurality of] microinstructions.

14. (Amended) The computer system [method] of claim 13, wherein the least two of a plurality of microinstructions are executed on separate execution units, but appear as though they were executed on a single execution unit.

15. (Amended) The computer system of claim 14, wherein the processor is further configured to emulate an instruction set by updating a flag based upon a result of the execution of the [at least two of the plurality of] microinstructions.

16. (Amended) The computer system of claim 15, wherein the processor is further configured to emulate[d] an instruction set by

[(a)] determining whether an exception occurs in the execution of any of the [at least two of a plurality of] microinstructions, [(b)] if an exception occurs, causing the exception to cancel all of the [at least two of a plurality of] microinstructions.

Claims 19 and 20 have been added as follows:

19. (New) The method of claim 1, wherein the step of issuing comprises forcing the microinstructions to issue simultaneously, in lockstep with each other, and wherein the step of canceling comprises canceling all of the plurality of microinstructions without regard to the relative ages of the microinstructions and without using a backoff mechanism.

20. (New) The method of claim 10, wherein the step of forcing the high-half and low-half operations to issue in parallel comprises causing the high-half and low-half operations to execute simultaneously in lockstep with each other, and wherein the step of flushing a result comprises canceling each of the high-half and low-half operations if an exception is taken in either the first or second FP unit.